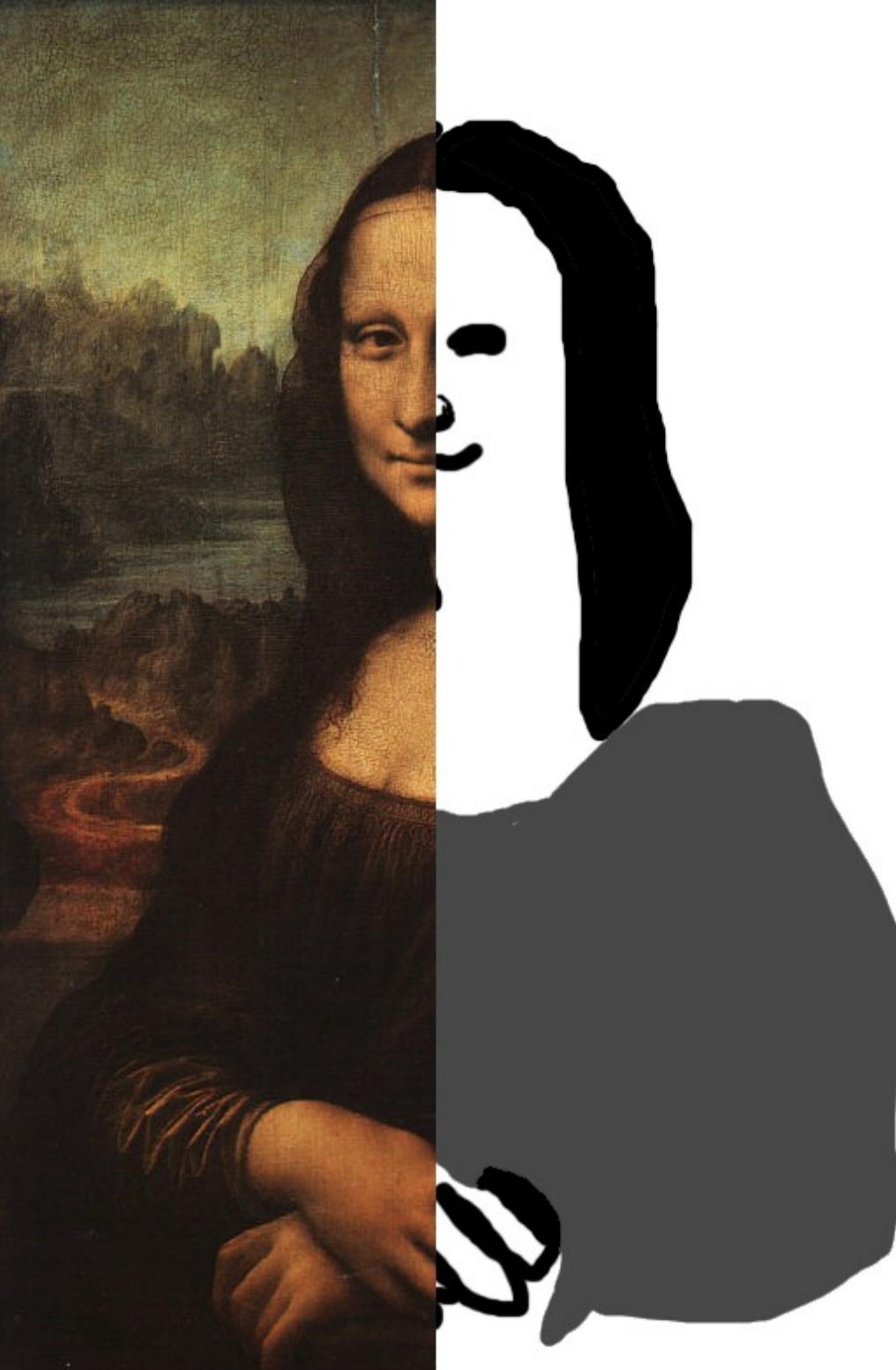




workshops
in creative coding

Theodoros
Papatheodorou

introduction



Computer Vision:

A set of algorithms that allow computers to understand images.

3D face recognition
by machines

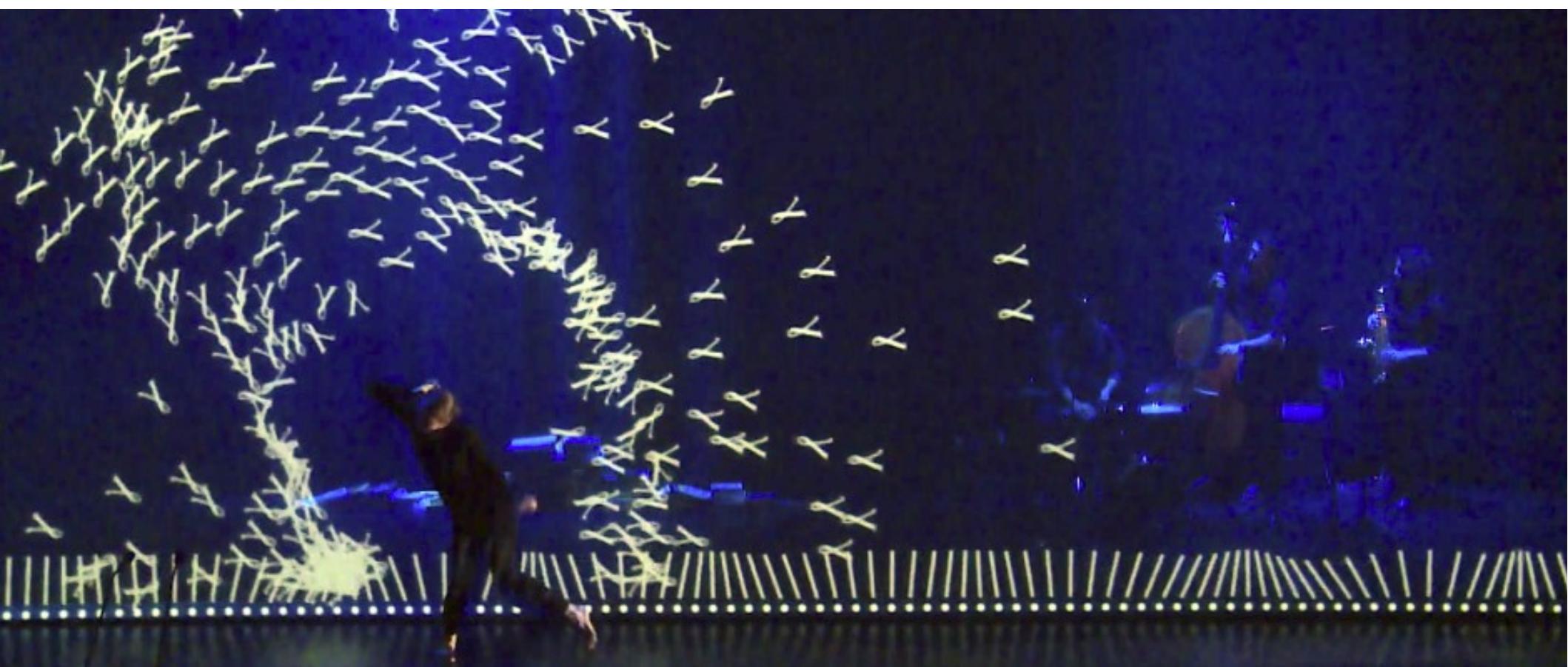




Athens School Of Fine Arts



interactive projections & computer vision in live performance



class purpose



- provide a fundamental understanding of code and modern computer literacy.
- introduce you to a range of techniques and practices for creating interactive audiovisual software in openFrameworks/C++
- develop your artistic practice
- assist you in delivering two working software projects

workshops in creative coding
v
programming for artists



class outline

- 1st term
 - syllabus
 - final project (video / website / source code)
- 2nd term
 - syllabus (with your input)
 - final project (video / website / source code)



bureaucratic

- email: t.papatheodorou@gold.ac.uk
- mailing list of MA/MFA: compArts@doc.gold.ac.uk
- mailing list of class: TBC
- class website: gitlab.doc.gold.ac.uk
- absence policy:
- grading: see class website on gitlab.doc.gold.ac.uk



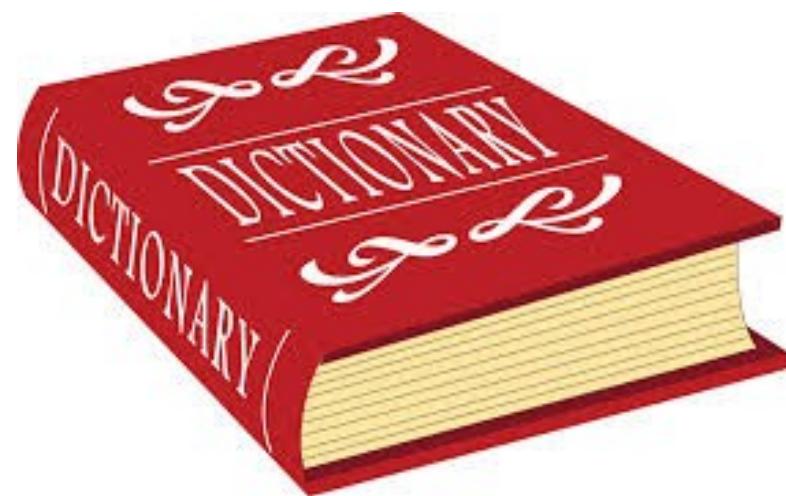
what is computational art?

“Could the artwork have been made without the use of a computer?”

“Does it take advantage of the computer’s unique capabilities?”

Roger Malina

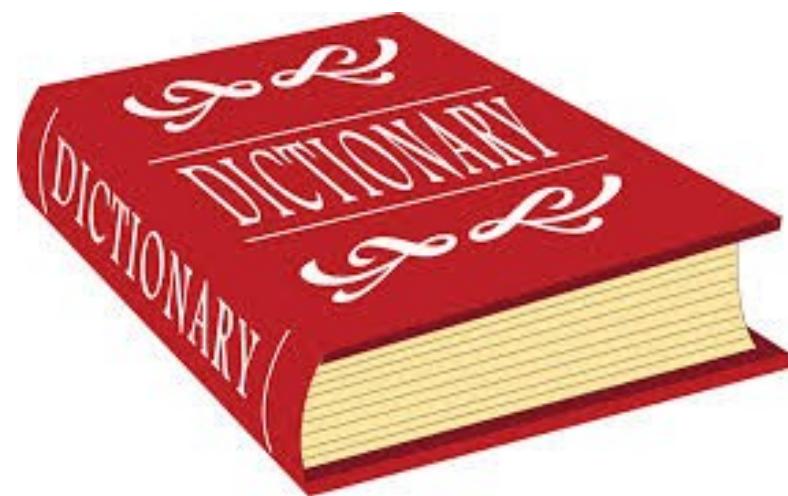
SIGGRAPH '89, Leonardo Special Edition



what is computational art?

“Does your work with the computer affect the direction of your results?”

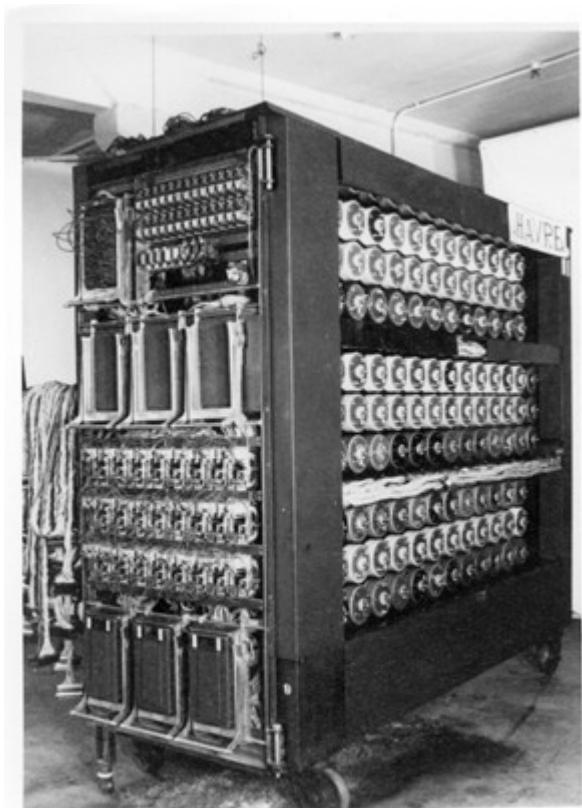
Aldo Giorgini – 1975
[source](#)



40s

code + arts

- programming: a difficult task
- designers + artists have different requirements than engineers and scientists
- C++/Java take years to master



"The world was at war. Complex calculations had to be done under time pressures not normally felt by mathematicians. It's unlikely that they gave even a passing thought to making computers user-friendly to people with softer styles than theirs."

- Seymour Papert

50-60s

code + arts

- artists experimented with themes related to software and system aesthetics
 - **Cybernetic Serendipity** exhibition (ICA, 1968)
 - Hans Haacke's **Visitor's Profile** (1971)



Cybernetic Serendipity

Serendipity

Serendipity

the faculty or making
happy chance discoveries of

Serendipity
means of control and communication machines
both human and electronic

An exhibition

In 1968, the Institute of Contemporary Arts
invited the artist and writer John Cage to
organise the course of this exhibition.
What was to become known as
the 'Year of Serendipity' was
born.

Computer generated graphics
computer programmed robots
computer music
computer poetry
computer games
computer simulations

and

other
serendipitous
manifestations

Institute
of Contemporary
Arts

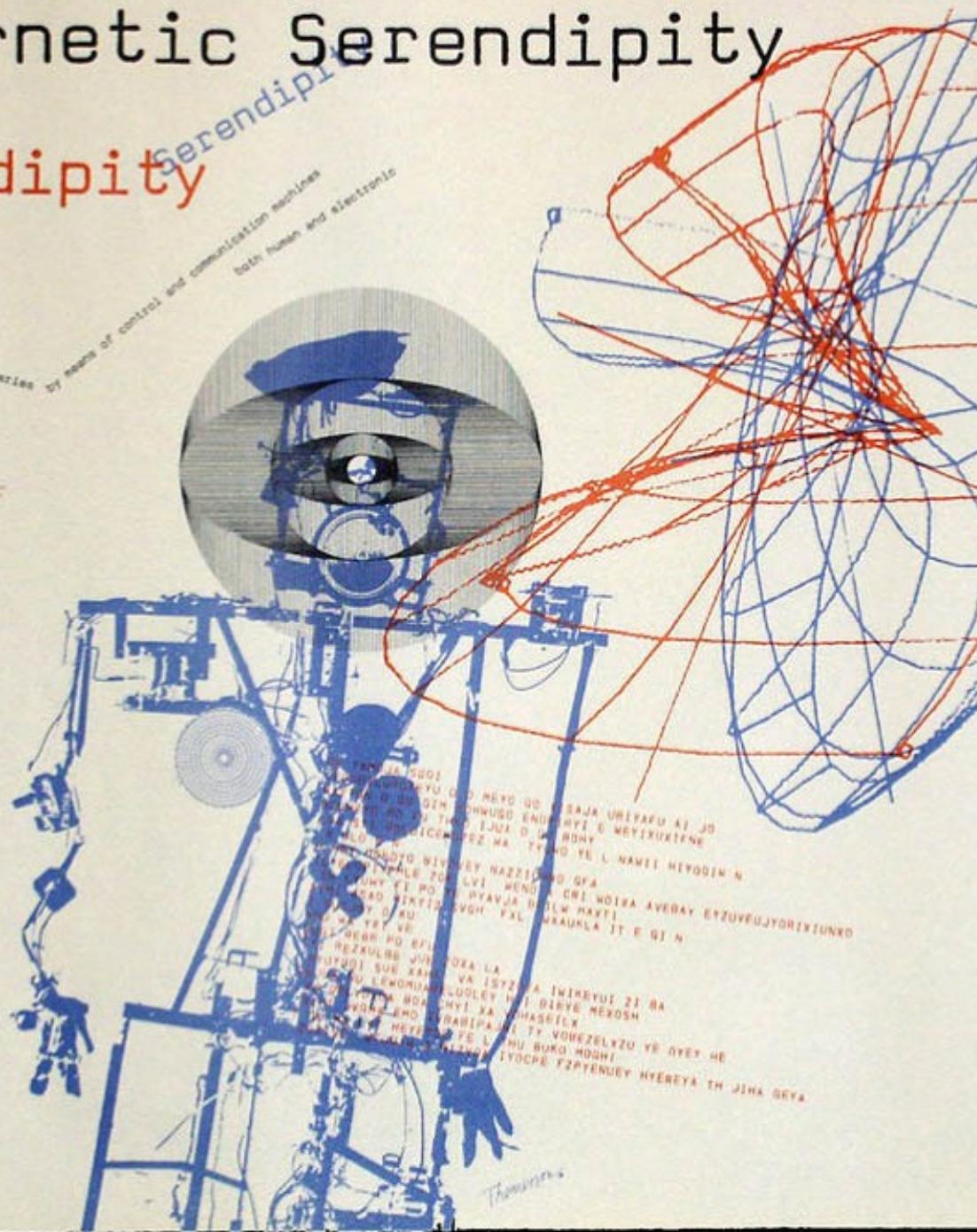
August 3 - October 26



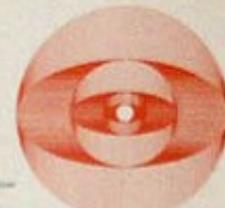
Institute of Contemporary Arts
Book House The Bell Tower 1968
August 3 - October 26

Galleries, Projections, Performances
August 3 - 10
Gardens, Garden
August 3 - 26

Admission by admission and reservation
for admission
entry £1.00



CYBERNETIC SERENDIPITY LECTURES



18
August 3 - October 26, 1968
"Using the ideas
of the Cybernetic Serendipity exhibition
at the Institute of Contemporary Arts
in New York, the MIT Center for the
Study of Man presents a series of lectures
on the subject of
Serendipity."

Thursday
August 8

Tuesday
August 13

Thursday
August 15

Tuesday
August 20

Thursday
August 27

Tuesday
September 3

Thursday
September 5

Tuesday
September 10

Thursday
September 12

Tuesday
September 19

Thursday
September 24

Thursday
September 26

Tuesday
October 1

Tuesday
October 8

Thursday
October 10

Thursday
October 17

Frank E. Hertzog
Actor in the Faculty of Architecture,
Professor, Massachusetts Institute of Technology
and the MIT Center for the
Study of Man.

Professor Herbert Ainsworth
Designer and manufacturer of robots
and the Director of the
Computer's Robot Service Bureau

Donald Glaser
Director of the Defense Nuclear Agency in New York
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

John Holloman
Professor of Mathematics
at the Institute of Computer Science
at the University of Waterloo
Waterloo, Ontario, Canada

Paul Morris
Mathematician, statistician and computer
science student of the Institute of Computer
Science, University of Waterloo

Michael S.A. Plataniotis
of the Department of Mathematics
and the Faculty of Computer Science
of the University of Waterloo
Waterloo, Ontario, Canada

Dr. Stephen Read
Professor of Psychology and Cybernetics
Director of Research at British Research Council
COMITEX in the UK and Ireland

Leatrice Rosen
Professor of Drama at the City College of New York
New York, New York
DIRECTOR OF THE CITY COLLEGE DRAMA

Paulo Ribeiro
Professor of Art and Design
of the Faculty of Fine Arts
and the Faculty of Architecture
of the University of São Paulo
São Paulo, Brazil

Professor Michael Ross
Professor of Mathematics, Linguistics, and Cybernetics
and the Director of the Institute of Computer Science
and the Institute of Mathematics and Physics
of the University of Waterloo

Professor A.J. Rosenberg
of the Institute of Computer Science in London
The Director of Computing to the Art of Computing
London, England

Professor John Salle
of the Department of Radiology Research Institute
DIRECTOR OF COMPUTERS

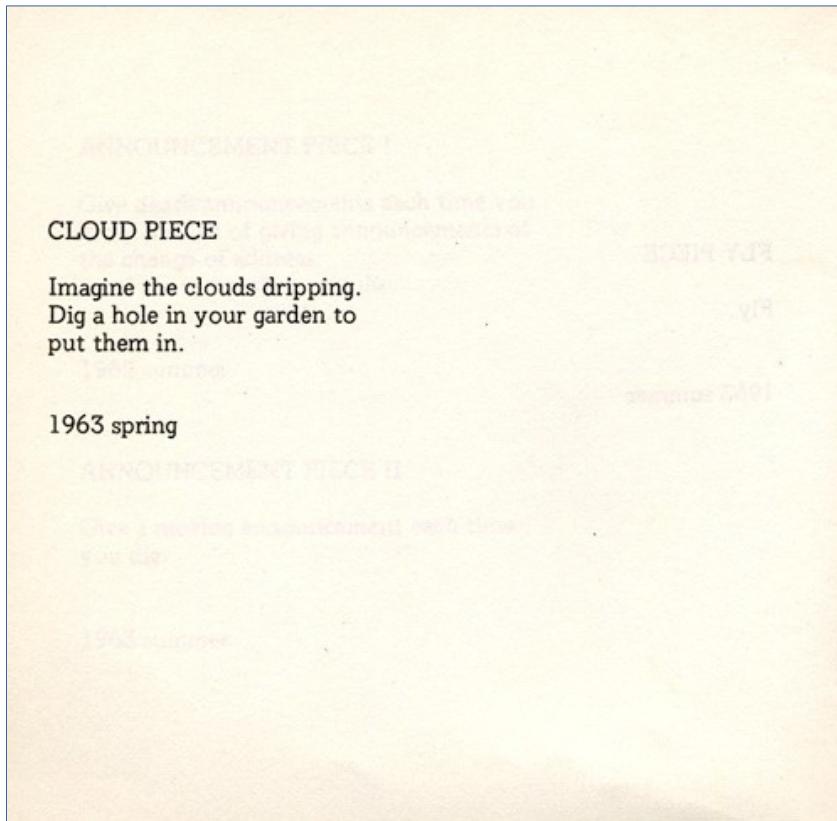
Matthew L. Tolson
Executive Director of Water Services
of the Faculty of Architecture
University of Waterloo, Waterloo, Ontario, Canada

Keith Wilson
Computer system designer and Research Director
of the Medical Computer Research Unit
YORK UNIVERSITY

50-60s cont.

code + arts

- process-based art
 - instructions and diagrams as a form of art



Yoko Ono (1963)

PROPOSAL FOR WALL DRAWING, INFORMATION SHOW

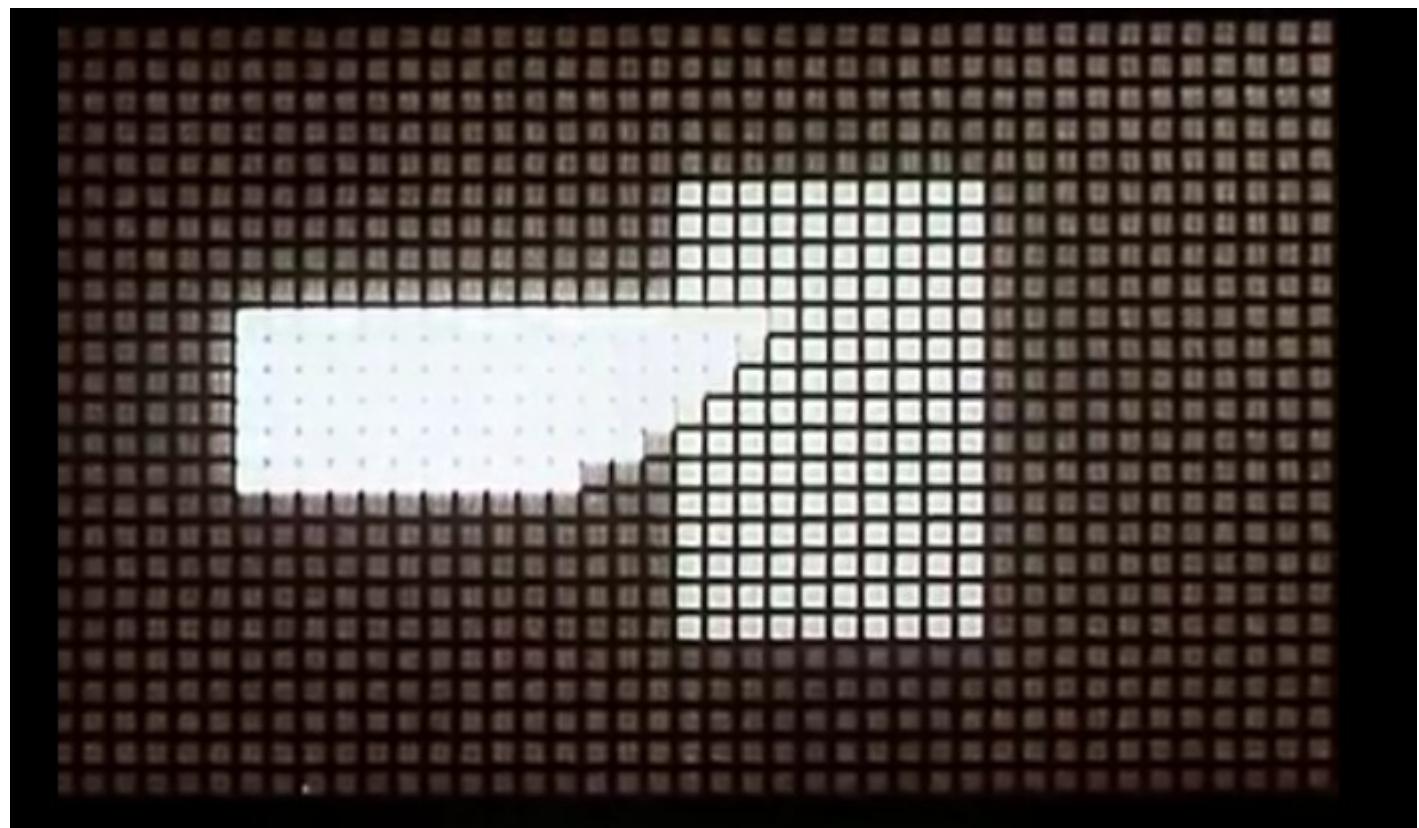
Within four adjacent squares,
each 4' by 4',
four draftsmen will be employed
at \$4.00/hour
for four hours a day
and for four days to draw straight lines
4 inches long
using four different colored pencils;
9H black, red, yellow and blue.
Each draftsmen will use the same color throughout
the four day period,
working on a different square each day.

Sol LeWitt (1970)

50-60s cont.

meanwhile the engineers...

BEFLIX (Bell Labs) – early computer graphics

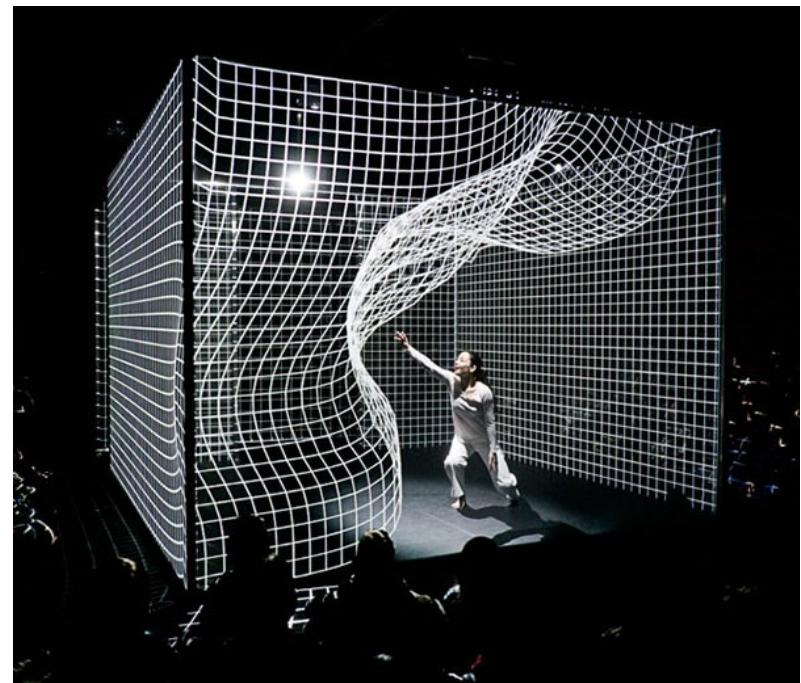


80s - today

code + arts



- proliferation of the personal computer
- programming reaches wider audience
- Lingo was developed
 - first language used by designers and artists up to early 90's
- code is used today not just for controlling pixels
 - controls elements of products
 - architecture
 - installation

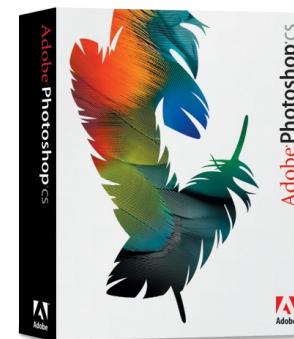


why code?

Tradition of artists creating their own tools as old as art itself.



Software revolution and the “Photoshop effect”



From paint to code



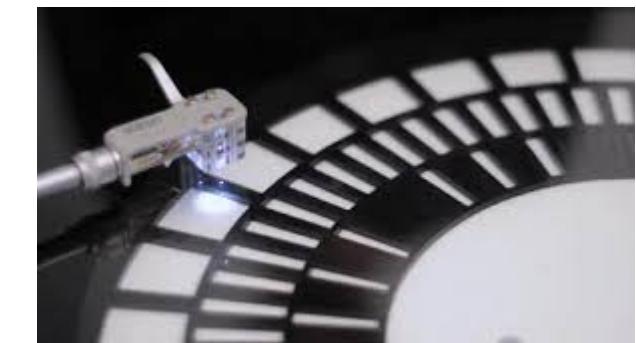
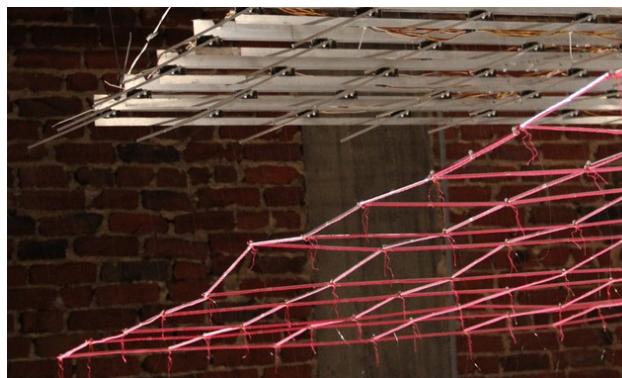
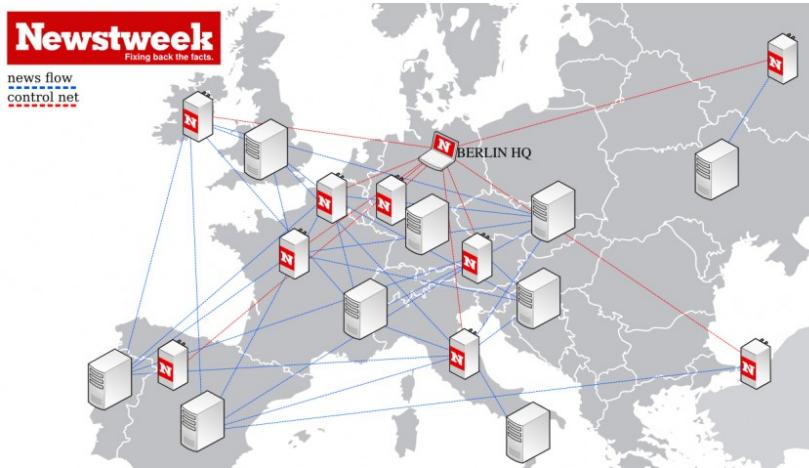


code + arts

conclusions

- observer vs. observed
- new modes of expression
- computer more than a projector
- non-linear narrative
- New ways of I/O
- exceed limits of proprietary packages
- gateway to new media
- computer becomes a medium, not a tool

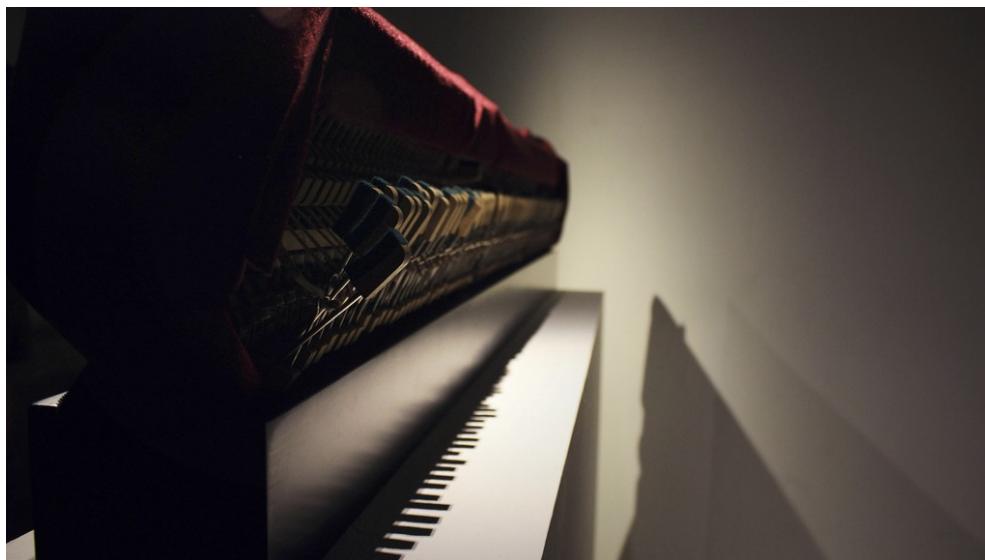
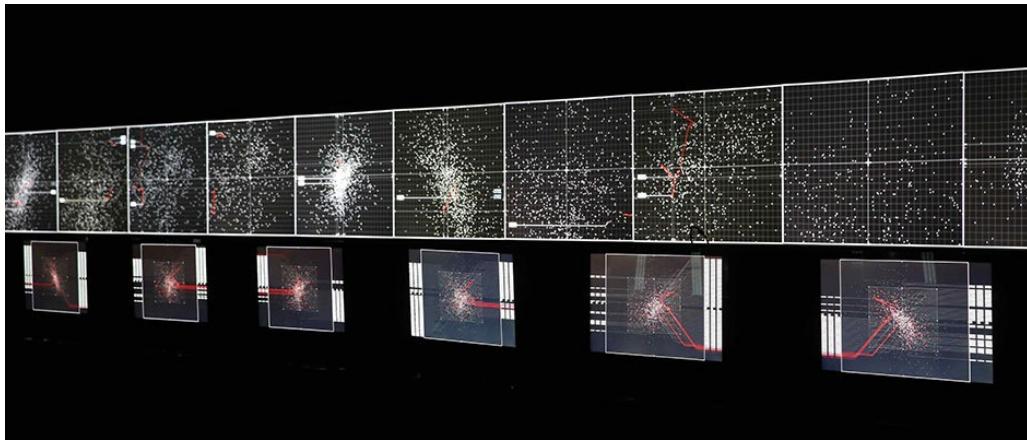
programming + art examples



n give her sorrow so much
ll be. FRIAR LAUREN
S Thou wrong'st it, more
is holy kiss. Exit JUL
oin'd my heart and Rome
FRIAR LAURENCE Ho
e serpents are; chain me
alone; Let not thy nurse
row'd likeness of shrunk
ight Shall Romeo bear t
and two Servingmen CA
use Ay, forsooth. CAPU
ell him of this: I'll have t

programming + art

examples (cont)



two volunteers for next time

- presentation of an art piece in 5'
- must be computational art
- among other things you must say:
 - who made it and when?
 - what technology was used?
 - how does it work? (describe its behavior)
 - what did you like in the artwork?
- for inspiration see links on class website





OpenFrameworks

**Free Software
Free Society**



a bit more history...

...a bit more history

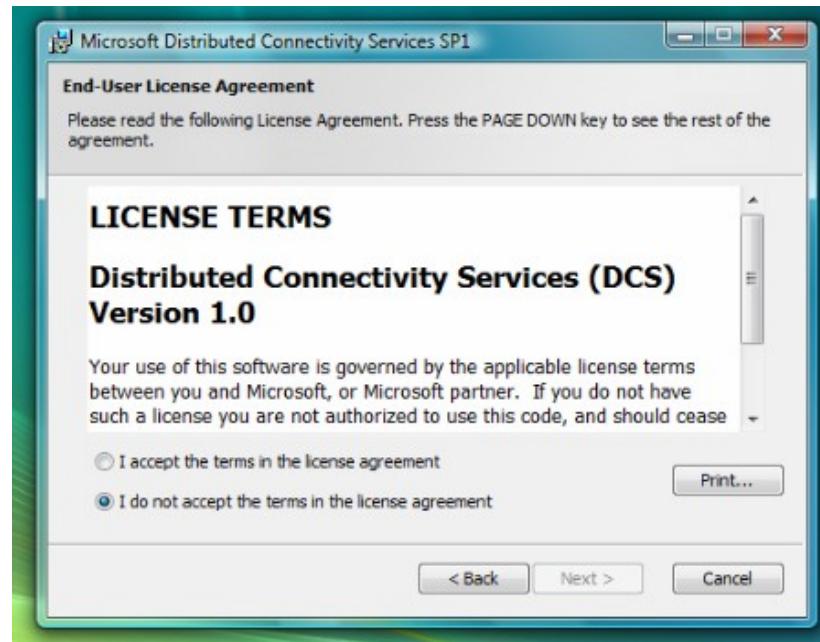


- Up until the late 70's users were able to study the source code of software, copy it and exchange it with each other
- But things were beginning to change since the end of the 60's:
 - increase in software costs
 - many users were asking for support
 - many users wanted to buy software separately
 - the software industry started competing with the hardware industry

a change...



- In the late 70's companies started distributing binary copies of their software
- Laws on intellectual property were extended to software
 - a software was protected just like a song or novel
 - they started selling software licenses and not the actual programs



Richard Stallman



free software

the basic freedoms

- **Freedom 0:** The freedom to run the program, for any purpose
- **Freedom 1:** The freedom to study how the program works, and change it so it does your computing as you wish. Access to the source code is a precondition for this.
- **Freedom 2:** The freedom to redistribute copies so you can help your neighbor
- **Freedom 3:** The freedom to distribute copies of your modified versions to others. By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this.

**FREE AS IN
FREEDOM**
**RICHARD STALLMAN'S
CRUSADE FOR FREE SOFTWARE**

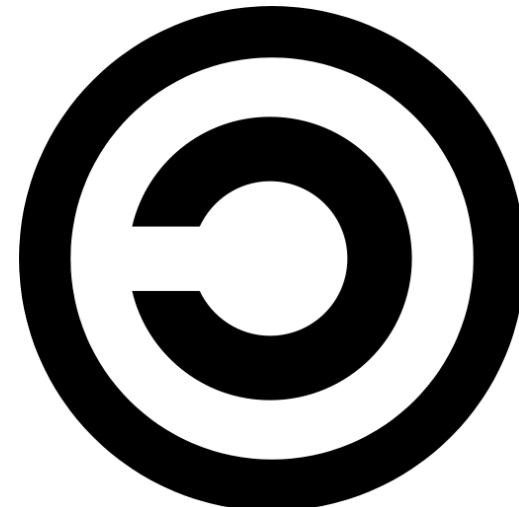


GNU General Public License

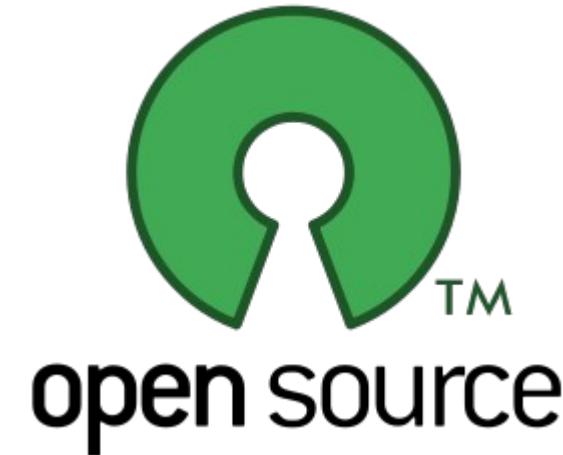
- publishes in 1989 the GPL software license in which apart from the aforementioned freedoms adds an original condition. *Copyleft*
- copyleft grants the freedom to use, modify and redistribute a intellectual creation...
- under the important condition that every copy or modified work will be distributed with the same license (granting the same freedoms)
- It's different from public domain



Free as in Freedom



free vs. open source



- open source: way of developing software
- free software: social movement



freedom and free software

- it's not just a good product
- it changes our way of thinking
- it contributes to the creation of a new social language
- new social and practical horizons open
- corporate controller of information (Google, Facebook et al.)

we must control the means of production



free software for artists

- 3D modelling software (Blender)
- 2D Animation (Ktoon, Synfig, Pencil)
- Video editing (KDEnlive, VirtualDub, Avidemux, Avisynth)
- Video players (VLC)
- Audio editing (Audacity)
- Processing / openFrameworks
- Pure Data / supercollider
- and many more here: <http://www.osalt.com/>



your turn

name

studies

relevant experience

interests

why are you here?

• programming experience





GitLab